DRONACHARYA
College of Engineering

INTELLIGENT SYSTEMS (CSE-303-F)

Section A

Informed Search

Heuristic Search Techniques

- Hill Climbing
- Best First Search
- A* Algorithm
- AO* Algorithm

Hill Climbing

- Uses simple heuristic function i.e. the amount of distance the node is from goal.
- ALGORITHM:

Step 1: Put the initial node on list START

Step 2: If (START is empty) or (START=GOAL) terminate search

Step 3: Remove the first node from START. Call this node a.

Step 4: If (a=GOAL) terminate search with success

Step 5: Else if node has a successor, generate all of them. Find out how far they are from goal node. Sort them by remaining distance from the goal and add them to the beginning of START.

Step 6: Go to Step 2.

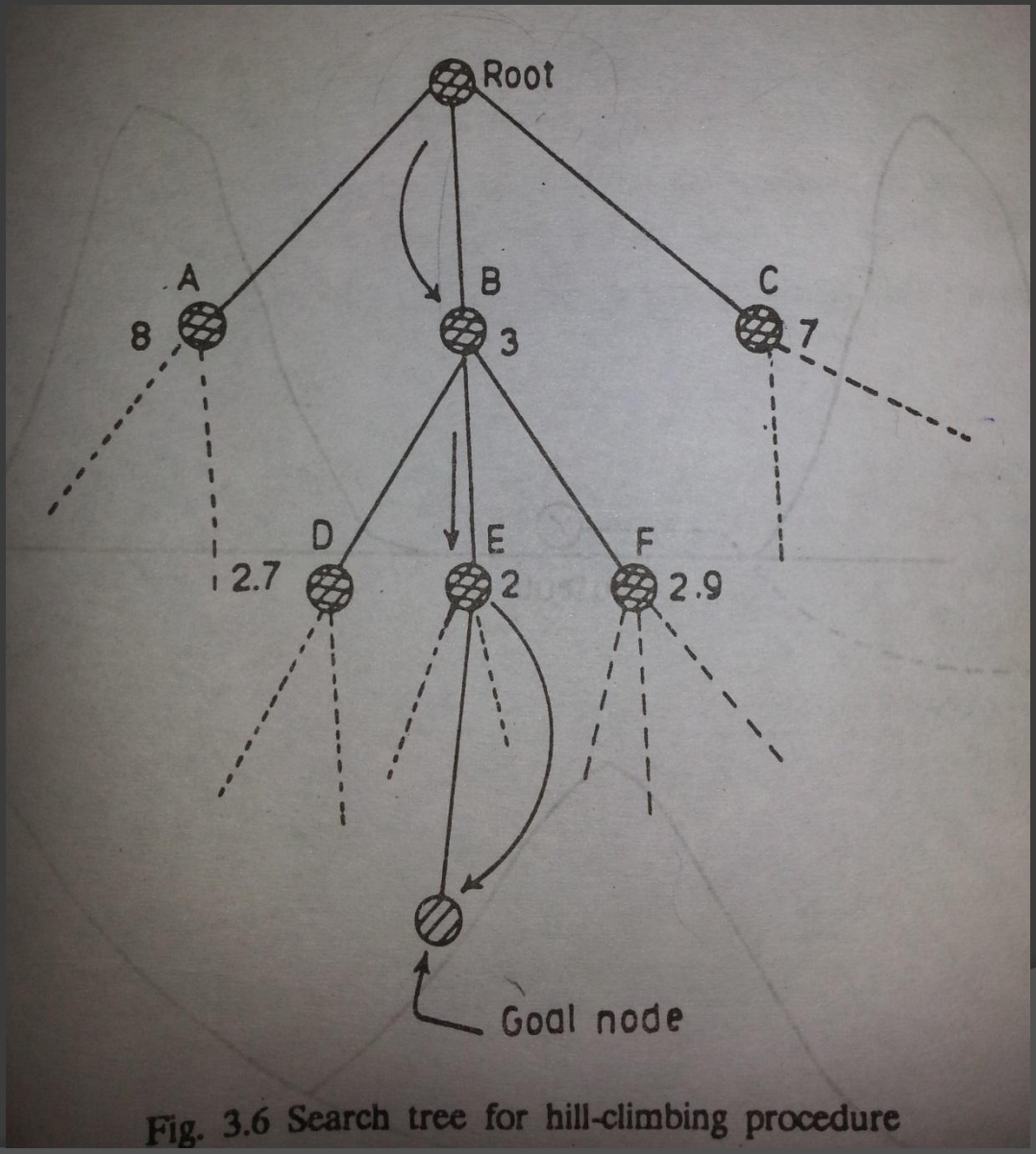


Fig. 3.6 Search tree for hill-climbing procedure

Problems with Hill Climbing

- Local Maximum : A state that is better than all its neighbors but is not better than some other states farther away.
- Plateau: A flat area in which set of neighboring states have same value.
- Ridge: Movement in any direction leads to same level o result.

Best First Search

- Heuristic function used is an evaluation function (indicates how far node is from goal node)
- Goal node has an evaluation function of zero.

Algorithm

Step 1: Put the initial node on list START

Step 2: If (START is empty) or (START=GOAL) terminate search

Step 3: Remove the first node from START. Call this node a.

Step 4: If (a=GOAL) terminate search with success

Step 5: Else if node has a successor, generate all of them. Find out how far they are from goal node. Sort all children generated so far by the remaining distance from goal.

Step 6: Go to Step 2.

Example

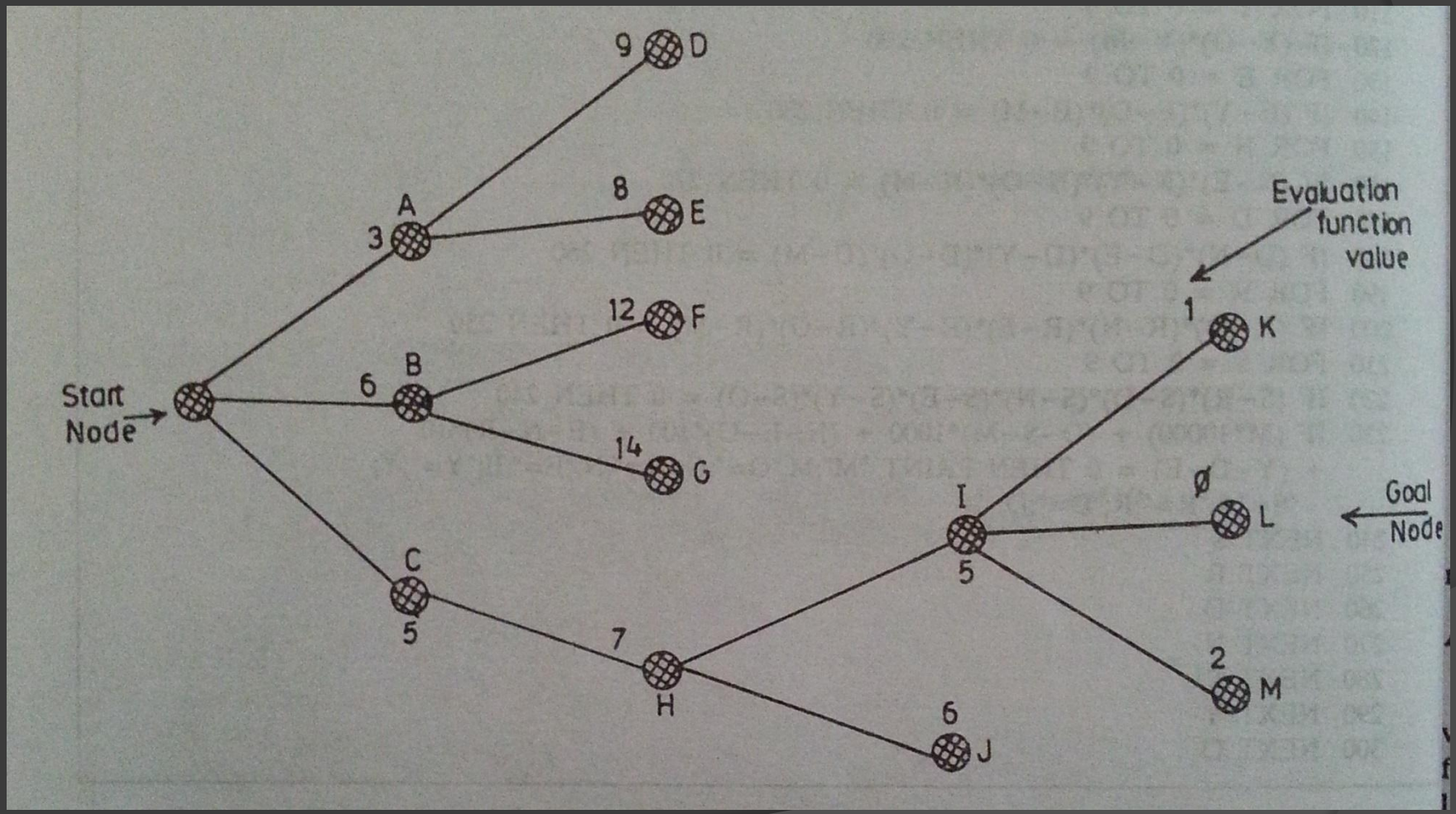

Start Node
A
3
B
6
C
5
9
D
8
E
12
F
14
G
7
H
6
J
I
5
1
K
ø
L
2
M
Evaluation function value
Goal Node

Table 3.1 Search process of best-first search

Step #	Node being expanded	Children	Available nodes	Node Chosen
1	S	(A : 3), (B : 6), (C : 5)	(A : 3), (B : 6), (C : 5)	(A : 3)
2	A	(D : 9), (E : 8),	(B : 6), (C : 5), (D : 9), (E : 8)	(C : 5)
3	C	(H : 7)	(B ; 6), (D : 9), (E : 8), (H : 7)	(B : 6)
4	B	(F : 12), (G : 14)	(D : 9), (E : 8), (H : 7) (F : 12), (G : 14)	(H : 7)
5	H	(I : 5), (J : 6)	(D : 9), (E : 8), (F : 12) (G : 14), (I : 5), (J : 6)	(I : 5)
6	I	(K : 1), (L : 0) (M : 2)	(D : 9), (E : 8), (F : 12) (G : 14), (J : 6), (K : 1) (L : 0), (M : 2)	Search stops as goal is reached

- There is only minor variation between hill climbing and best first search
- In first approach, the children of first node are being generated
- But in second approach, we have to sort entire list to identify next node to be expanded.

A* Algorithm

- Uses evaluation function and cost functions as heuristic.
- Sum of evaluation function and cost leading to goal state is called fitness value.

Algorithm

Step 1: Put the initial node on list START

Step 2: If (START is empty) or (START=GOAL) terminate search

Step 3: Remove the first node from START. Call this node a.

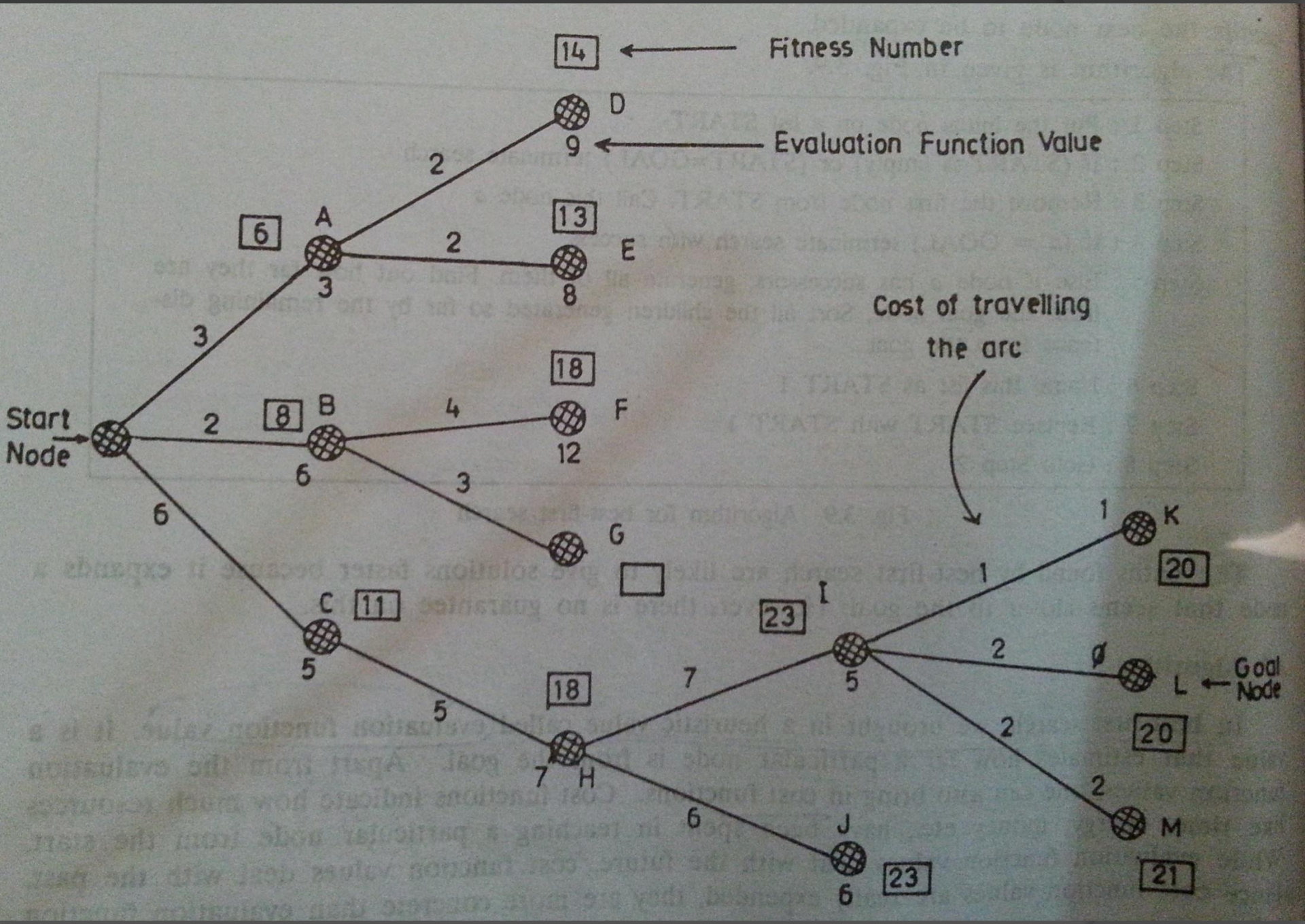
Step 4: If (a=GOAL) terminate search with success

Step 5: Else if node has a successor, generate all of them. Estimate the fitness number of the successors by totaling the evaluation function value and cost function value. Sort the list by fitness number.

Step 6: Name the new list as START 1.

Step 7: Replace start with START 1.

Step 8: Go to Step 2.


14
Fitness Number
D
9
Evaluation Function Value
2
6
A
13
2
E
3
8
Cost of travelling
the arc
3
18
Start
Node
2
8
B
4
F
12
6
3
6
G
1
K
20
C
11
I
23
1
5
2
ø
L
Goal
Node
18
7
5
20
7
H
2
2
6
J
M
6
23
21